

# (12) UK Patent Application (19) GB (11) 2 354 848 (13) A

(43) Date of A Publication 04.04.2001

(21) Application No 9922818.1

(22) Date of Filing 28.09.1999

(71) Applicant(s)

**International Business Machines Corporation**  
(Incorporated in USA - New York)  
Armonk, New York 10504, United States of America

(72) Inventor(s)

Stephen J Todd  
Malcolm D Ayres  
Peter A Lambros  
Timothy Holloway  
Marc-Thomas Schmidt

(74) Agent and/or Address for Service

Kevin J Fournier  
IBM United Kingdom Limited, Intellectual Property  
Department, Hursley Park, WINCHESTER, Hampshire,  
SO21 2JN, United Kingdom

(51) INT CL<sup>7</sup>

G06F 9/46 17/30

(52) UK CL (Edition S)

G4A AFGDC APX AUDB

(56) Documents Cited

EP 0769591 A1 WO 91/03017 A1 US 4815030 A

(58) Field of Search

UK CL (Edition R) G4A AFGDC APX AUDB  
INT CL<sup>7</sup> G06F

(54) Abstract Title

**Publish/subscribe data processing with subscriber requested messageflow for customised message processing**

(57) A message broker data processing method includes steps of: receiving 525 published messages on a topic from a publisher application 31; processing the received messages via a messageflow comprising at least one data processing node 521; distributing the processed messages to subscriber applications 33,34; receiving a subscription request from a subscriber application, the subscription request including an indication of a data processing node e.g. 526 which the subscriber application would like to add to the messageflow; and in response to the indication in the subscription request, adding the data processing node to the messageflow.

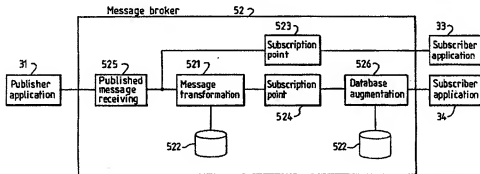
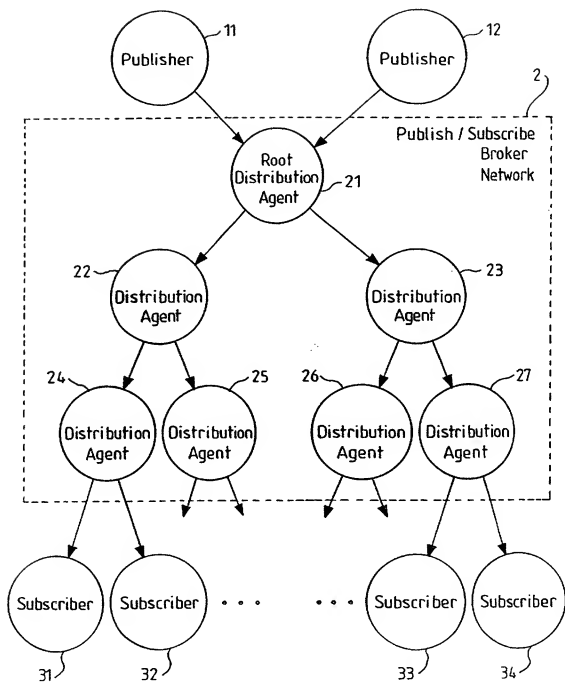
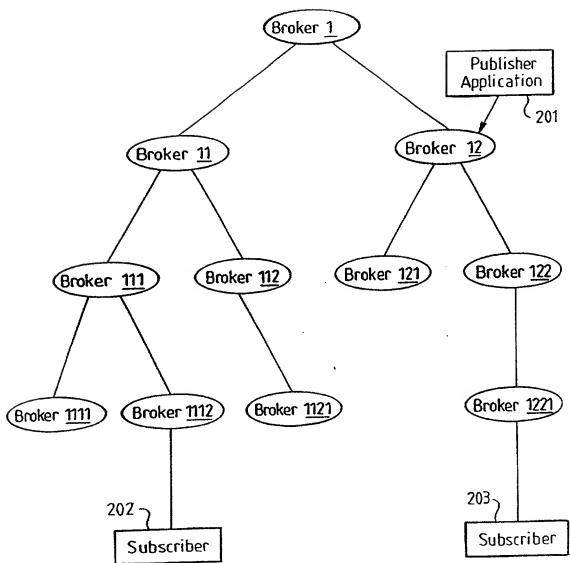
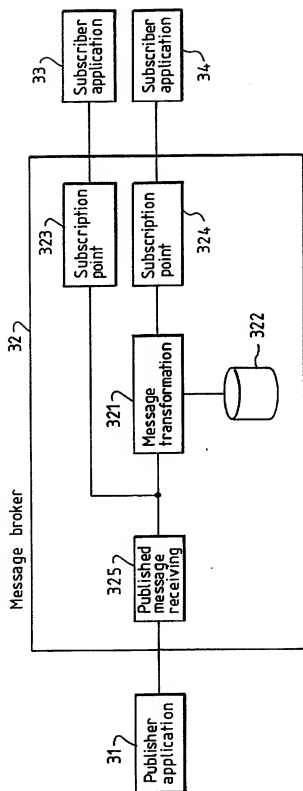


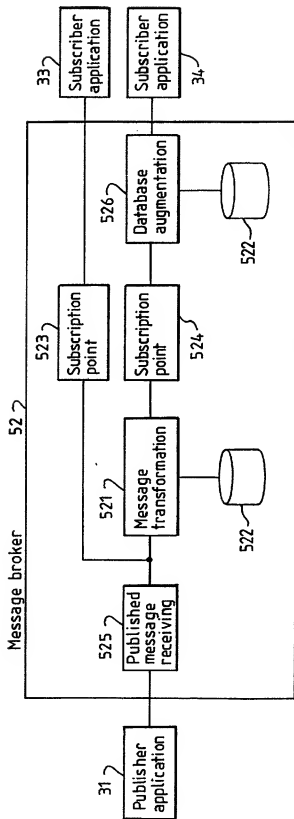
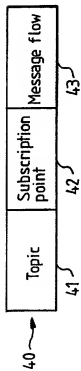
FIG. 5

GB 2 354 848 A

**FIG. 1**

FIG. 2

FIG. 3



PUBLISH/SUBSCRIBE DATA PROCESSING WITH SUBSCRIBER REQUESTED  
MESSAGEFLOW FOR CUSTOMISED MESSAGE PROCESSING

5

Field of the Invention

10 The present invention relates to the field of data processing and more specifically to event notification data processing which distributes event messages from suppliers (called, hereinafter, "publishers") of data messages to consumers (called, hereinafter "subscribers") of such messages. While there are many different types of known event notification systems, the subsequent discussion will describe the  
15 publish/subscribe event notification system as it is one of the most common.

Background of the Invention

20 Publish/subscribe data processing systems (and event notification systems in general) have become very popular in recent years as a way of distributing data messages (events) from publishing computers to subscribing computers. The increasing popularity of the Internet, which has connected a wide variety of computers all over the world, has helped  
25 to make such publish/subscribe systems even more popular. Using the Internet, a World Wide Web browser application (the term "application" or "process" refers to a software program, or portion thereof, running on a computer) can be used in conjunction with the publisher or subscriber in order to graphically display messages. Such systems are especially  
30 useful where data supplied by a publisher is constantly changing and a large number of subscribers needs to be quickly updated with the latest data. Perhaps the best example of where this is useful is in the distribution of stock market data.

35 In such systems, publisher applications of data messages do not need to know the identity or location of the subscriber applications which will receive the messages. The publishers need only connect to a publish/subscribe distribution agent process, which is included in a group of such processes making up a broker network, and send messages to  
40 the distribution agent process, specifying the subject of the message to the distribution agent process. The distribution agent process then distributes the published messages to subscriber applications which have previously indicated to the broker network that they would like to receive data messages on particular subjects. Thus, the subscribers also  
45 do not need to know the identity or location of the publishers. The subscribers need only connect to a distribution agent process.

One such publish/subscribe system which is currently in use, and which has been developed by the Transarc Corp. (a wholly owned subsidiary of the assignee of the present patent application, IBM Corp.) is shown in Fig. 1. Publishers 11 and 12 connect to the publish/subscribe broker network 2 and send published messages to broker network 2 which distributes the messages to subscribers 31, 32, 33, 34. Publishers 11 and 12, which are data processing applications which output data messages, connect to broker network 2 using the well known inter-application data connection protocol known as remote procedure call (or RPC) (other well known protocols, such as asynchronous message queuing protocols, can also be used). Each publisher application could be running on a separate machine, alternatively, a single machine could be running a plurality of publisher applications. The broker network 2 is made up of a plurality of distribution agents (21 through 27) which are connected in a hierarchical fashion which will be described below as a "tree structure". These distribution agents, each of which could be running on a separate machine, are data processing applications which distribute data messages through the broker network 2 from publishers to subscribers. Subscriber applications 31, 32, 33 and 34 connect to the broker network 2 via RPC in order to receive published messages.

Publishers 11 and 12 first connect via RPC directly to a root distribution agent 21 which in turn connects via RPC to second level distribution agents 22 and 23 which in turn connect via RPC to third level distribution agents 24, 25, 26 and 27 (also known as "leaf distribution agents" since they are the final distribution agents in the tree structure). Each distribution agent could be running on its own machine, or alternatively, groups of distribution agents could be running on the same machine. The leaf distribution agents connect via RPC to subscriber applications 31 through 34, each of which could be running on its own machine.

In order to allow the broker network 2 to determine which published messages should be sent to which subscribers, publishers provide the root distribution agent 21 with the name of a distribution stream for each published message. A distribution stream (called hereinafter a "stream") is an ordered sequence of messages having a name (e.g., "stock" for a stream of stock market quotes) to distinguish the stream from other streams (this is known as "topic based" publish/subscribe, another well known model is called "content based publish/subscribe which involves matching publishers and subscribers by the content of the messages rather than by the topic). Likewise, subscribers provide the leaf distribution agents 31 through 34 with the name of the streams to which they would like to subscribe. In this way, the broker network 2 keeps track of which subscribers are interested in which streams so that when publishers publish messages to such streams, the messages can be distributed to the

corresponding subscribers. Subscribers are also allowed to provide filter expressions to the broker network in order to limit the messages which will be received on a particular stream (e.g., a subscriber 31 interested in only IBM stock quotes could subscribe to the stream "stock" by making an RPC call to leaf distribution agent 24 and include a filter expression stating that only messages on the "stock" stream relating to IBM stock should be sent to subscriber 31).

The above-described publish/subscribe architecture provides the advantage of central co-ordination of all published messages, since all publishers must connect to the same distribution agent (the root) in order to publish a message to the broker network. For example, total ordering of published messages throughout the broker network is greatly facilitated, since the root can easily assign sequence numbers to each published message on a stream. However, this architecture also has the disadvantage of publisher inflexibility, since each publisher is constrained to publishing from the single root distribution agent, even when it would be much easier for a publisher to connect to a closer distribution agent.

In the Fig. 1, a publisher application 11, running on one computer, is, for example, a supplier of live stock market data quotes. That is, publisher application 11 provides frequent messages stating the present value of share prices. In this example, publisher application 11 is publishing messages on a stream called "stock" which has already been configured in the broker network 2. As is well known, when publisher 11 wishes to publish a stock quote message to stream "stock", publisher 11 makes an RPC call to the root distribution agent 11 which is at the top level of the broker network tree structure. In this example, subscriber application 32, running on another computer, has sent a subscription request via an RPC call to leaf distribution agent 24, which is at the bottom level of the tree structure, indicating that subscriber 32 would like to subscribe to stream "stock".

Thus, whenever publisher 11 publishes a data message to stream "stock" the distribution tree structure of broker network 2 channels the message down through the root distribution agent 21, through any intermediary distribution agents (e.g., 22 in the example of Fig. 1) and through the leaf distribution agent 24 to the subscriber 32. This involves a series of RPC calls being made between each successive circle in the diagram of Fig. 1 connecting publisher 11 and subscriber 32 (i.e., 11 to 21, 21 to 22, 22 to 24 and 24 to 32).

Figure 2 shows a different publish/subscribe architecture where publisher applications can publish messages to the broker network by directly communicating with any one of a plurality of distribution agents

(brokers). For example, publisher application 201 is shown communicating directly with Broker 12. There is no requirement in this architecture that all publisher applications communicate directly with a top (or root) distribution agent. Publisher application 201 can potentially communicate directly with any of the distribution agents shown in Fig 2, in the described examples below it will be shown communicating directly with Broker 12.

Subscriber applications 202 and 203 would like to receive messages on the stream/topic that publisher application 201 is publishing on. Thus, subscriber applications 202 and 203 communicate directly with Brokers 1112 and 1221, respectively, to provide subscription data thereto informing the broker hierarchy of their desire to receive such published messages. Since the publisher application 201 is allowed to communicate directly with any of a plurality of distribution agents, the subscription data entered by the subscriber applications must be propagated throughout the broker network to each Broker shown in Fig. 2. This way, no matter which distribution agent the publisher application 201 happens to communicate directly with, the published messages will be able to be routed to the subscriber applications 202 and 203.

Publish/subscribe broker systems have commonly been integrated into multi-function message broker systems which are used to inter-connect applications which may be on heterogeneous platforms and may use different message formats. For example, Saga Software of Reston, Virginia (USA) ([www.sagasoftware.com](http://www.sagasoftware.com)) have such a message broker product called "Sagavista" (a trademark of Saga Software). Further, Tibco Software Inc. of Palo Alto, California (USA) ([www.tibco.com](http://www.tibco.com)) also have such a message broker called "TIB/Message Broker" (both "TIB" and "TIB/Message Broker" are trademarks of Tibco). In these multi-function message brokers, a set of pluggable data processing nodes is provided, with each node being dedicated to a specific data processing task, such as message format transformation, publish/subscribe message distribution, and a rules engine for deciding (based on a plurality of predefined rules) where an incoming message should be routed.

In these multi-function message broker products, when a subscriber application registers a subscription request with the broker, the subscriber application sends the subscription request to a publish/subscribe broker node specifying the topic of the desired subscription. The publish/subscribe broker node (usually in cooperation with a plurality of other such publish/subscribe broker nodes) then ensures that any published messages on that topic are sent to the subscriber application. While the multi-function message broker provides a plurality of messageflow nodes which carry out various data processing operations, different subscribers may wish to receive publications on a

given topic with the messages having passed through a customized version of a messageflow. For example, a subscriber may not only want to receive published messages on the topic "IBM stock" but also would like the broker to add an additional field to any such messages to inform the subscriber of how many shares of IBM stock that subscriber currently owns. This would require the addition of a database augmentation node which would join a published message with an output of a database which holds data concerning how many shares of IBM stock subscribers currently own.

In order to accommodate such customization desires of various subscribers, a systems administrator of the message broker would have to modify the messageflow nodes to add the respective customized nodes as requested by the subscribers. This can become burdensome on the systems administrator, especially as the number of subscribers requiring customization increases to very high levels.

#### Summary of the Invention

According to one aspect, the present invention provides a message broker data processing apparatus including: a unit for receiving published messages on a topic from a publisher application; a unit for processing the received messages; and a unit for distributing the processed messages to subscriber applications; where the apparatus further includes: a unit for receiving a subscription request from a subscriber application, the subscription request including an indication of a data processing node which the subscriber application would like to add to the apparatus; and a unit for, in response to the indication in the subscription request, adding the data processing node to the apparatus.

According to a second aspect, the present invention provides a data processing method of carrying out the functionality recited above with respect to the first aspect.

According to a third aspect, the present invention provides a computer readable storage medium having a computer program stored on it which, when executed on a computer, carries out the functionality of data processing method of the second aspect of the invention.

Thus, the present invention provides a message broker having a publish/subscribe capability where subscribers are given the ability to request that the messageflow data processing nodes of the message broker be modified by the addition of nodes. Subscribers can thus initiate a customization of the messageflow themselves without having to involve the systems administrator. The message broker will react to the subscriber's

customization request and modify the messageflow without involving the systems administrator.

#### Brief Description of the Drawings

The invention will be better understood by referring to the detailed description of the preferred embodiments which will now be described in conjunction with the following drawing figures:

Figure 1 is a block diagram showing a first architecture of a publish/subscribe data processing system to which the preferred embodiment of the present invention can be advantageously applied;

Figure 2 is a block diagram showing a second architecture of a publish/subscribe data processing system to which the preferred embodiment of the present invention can be advantageously applied; and

Fig. 3 is a block diagram showing an exemplary message broker to which a preferred embodiment of the present invention can be advantageously applied;

Fig. 4 shows the format of a subscription request according to a preferred embodiment of the present invention;

Fig. 5 is a block diagram showing an exemplary message broker according to a preferred embodiment of the present invention.

#### Detailed Description of the Preferred Embodiments

In Fig. 3 a message broker 32 receives published messages on a topic called "IBM stock" from a publisher application 31 (which is an application running at a major stock exchange) and distributes such published messages to subscriber applications 33 and 34 (which are two different stock broking agencies), each of which have previously registered subscriptions to the topic "IBM stock". In this example, the publisher application, broker and subscriber applications are all running on separate machines (and are thus interconnected via a network which is not shown in Fig. 3). In other embodiments, however, two or more of the applications (e.g., the publisher and the broker) could be running on the same machine. Further, as was explained above, the broker 32 is most likely running on a plurality of machines.

When the published messages are received into the message broker 32 at a published message receiving data processing node 325, they are sent to a message transformation data processing node 321 which performs the function of transforming the format of the published messages so that the

IBM stock prices, which are originally published in US dollars, are converted to UK pounds. The message transformation node 321 accesses local storage 322 in order to determine the current exchange rate of US dollars to UK pounds (this exchange rate is updated at the beginning of every business day). After having their US dollar amounts converted to UK pounds, the messages are output from the message transformation node 321 and received at a subscription point processing node 324.

A subscription point data processing node (or "subscription point" for short) is an instance of a publish/subscribe matching engine which performs the function of looking at the topics in previously received subscription requests (received from subscribers) and determining whether the topic in an incoming message (just received from a publisher application) matches the topic of any of the previously received subscription requests. For any subscriptions that match, the subscription point data processing node distributes the published message to the subscriber application(s) which had entered the subscription requests.

Back to Fig. 3, the subscription point processing node 324 determines (e.g., by accessing local storage 322) that subscriber application 34 has previously entered a subscription on the topic "IBM stock". Thus, subscription point processing node 324 distributes the published messages to subscriber application 34.

Upon entering the broker 32, published messages from publisher 31 also take another path bypassing the message transformation data processing node 321. Specifically, the published messages also are sent directly to subscription point data processing node 323 (which is separate from subscription point data processing node 324 discussed earlier). Subscription point data processing node 323 then performs a publish/subscribe topic matching operation and determines that subscriber application 33 has previously entered a subscription request to the topic "IBM stock". Thus, subscription point processing node 323 distributes the published messages to subscriber application 33.

When subscriber application 33 registered its subscription to the topic "IBM stock" with the broker 32, subscriber application 33 provided the broker not only with the name of the topic "IBM stock" but also with the identity of subscription point 323 as the desired subscription point in the message flow of data processing nodes within the broker 32. This informs the broker 32 that subscription point 323 should be used to satisfy subscriber application 33's subscription request. Subscriber application 33 has selected subscription point 323 because subscriber application 33 is being run by a stock broker agency located in the United States and thus subscriber application 33 desires to receive

publications on topic "IBM stock" with the stock prices quoted in US dollars. Because the message flow through the data processing nodes of message broker 32 for subscription point 323 bypasses the message transformation node 321, the original format (stock prices in US dollars) of the published messages from publisher application 31 is maintained and the subscriber application 33 receives such published messages without message transformation.

On the other hand, when subscriber application 34 registered its subscription to the topic "IBM stock" with the broker 32, subscriber application 34 provided the broker not only with the name of the topic "IBM stock" but also with the identity of subscription point 324 as the desired subscription point in the message flow of data processing nodes within the broker 32. This informs the broker 32 that subscription point 324 should be used to satisfy subscriber application 34's subscription request. Subscriber application 34 has selected subscription point 324 because subscriber application 34 is being run by a stock broker agency located in the United Kingdom and thus subscriber application 34 desires to receive publications on topic "IBM stock" with the stock prices quoted in UK pounds. Because the message flow through the data processing nodes of message broker 32 for subscription point 324 includes the message transformation node 321, the original format (stock prices in US dollars) of the published messages from publisher application 31 is converted into UK pounds and the subscriber application 34 receives such published messages after having had the format of such messages transformed.

Thus, by the use of a plurality (two in Fig. 3) of subscription point data processing nodes in a message broker, subscriber applications can be provided with published messages in a message format selected by the subscriber without having to use different topics (the topic "IBM stock" is the same for both subscription points 323 and 324). This allows access control to be easily carried out on a topic basis. For example, the broker can perform a security measure on both subscriber applications 33 and 34 by simply checking whether the requested topic "IBM stock" of their subscription requests is a topic which has previously been determined as acceptable for subscribers 33 and 34 from a security standpoint.

The use of subscription points in message flows through the broker is not limited to message format transformation. Such subscription points could also be used to implement different filters on subscription topics. For example, message transformation data processing node 321 could be replaced with a topic filtering data processing node which carries out the function of filtering any published messages on topic "IBM stock" where the stock price is greater than 100 US dollars per share. That is, the filtering data processing node examines the stock

price in each published message on topic "IBM stock" and only if the stock price is greater than 100 dollars per share is the published message allowed to pass through the filtering node. The published messages that make it through the filtering node are then sent to subscription point data processing node 324 which performs a topic matching function (as described above) and distributes the published messages to subscriber application 34. In this case, subscriber application 34 is being run by a US-based stock broker agency which is only concerned with messages concerning IBM stock if the IBM stock price per share is greater than 100 US dollars.

According to a preferred embodiment of the present invention, when a subscriber application is registering a new subscription request 40 (see Fig. 4) to the broker, the subscriber application (e.g., 34 of Fig. 3) adds an additional field 43 called a "messageflow" field to the subscription request in addition to the fields 41 and 42 which indicate the desired topic and subscription point, respectively, of the subscription being requested. In this "messageflow" field 43, the subscriber application 34 inserts an indication of an additional messageflow data processing node which the subscriber would like added to the series of messageflow nodes 325, 321 and 324 of Fig. 3 in order to further customize the series of messageflow nodes.

For example, when subscriber 34 registers its subscription request 40 to the broker, subscriber 34 provides an indication of the topic "IBM stock" in field 41, an indication of subscription point 324 in field 42 and an indication in field 43 that subscriber 34 would like to add a database augmentation node which joins published messages with an output of a database which holds data concerning how many shares of IBM stock subscriber 34 currently owns. In response to receiving this subscription request, the broker 52 (see Fig. 5) adds database augmentation node 526 after the subscription point 524 in the series of messageflow nodes 525, 521, 524, 526. Any of a set of standard messageflow nodes (e.g., transformation, database augmentation, reformatting, etc.) recognized by the broker 52 could be added in this fashion.

In operation of the message broker according to the example provided above, published messages on topic "IBM stock" are received into the broker at the published message receiving data processing node 525, undergo message transformation at node 521 (US dollar amounts converted to UK pounds) and are then received at subscription point 524 for publish/subscribe matching purposes (all as discussed above). Before being forwarded to matched subscriber application 34, the database augmentation node 526 joins the published messages with the output of database 522 which holds the amount of shares of IBM stock currently

owned by subscriber 34. The so-augmented published messages are then forwarded to subscriber application 34.

5 A subscriber application 34 can request that a plurality of different messageflow nodes be added to the end of a subscription point. Further, the group of such added nodes can take input from a plurality of subscription points (e.g., 523 and 524).

10 A system may include multiple brokers each of which provides an equivalent set of multiple subscription points. These brokers are arranged as described earlier for publish/subscribe systems so that a subscriber subscribing to a particular subscription point receives the same set of messages regardless of the broker to which the subscription is made. Each broker subscribes to the subscription points of other  
15 brokers on behalf of client subscribers. Resulting interbroker messages all flow multiplexed onto a single communication channel (e.g. MQSeries queue) between two given connected brokers, regardless of the subscription point to which the message belongs. The source broker tags the interbroker message with header information indicating the  
20 subscription point to which the message belongs. The receiving broker uses the header tag to publish the messages to all subscribers at the given equivalent subscription point on the receiving broker.

25 While the preferred embodiment has been described with reference to the subscription points illustrated in Fig. 3, the invention can also be applied to other multi-function broker configurations which do not use subscription points (e.g., configurations where publisher applications publish the same messages on different topics in order to accommodate messages in different formats, and configurations where the broker  
30 modifies the topics to accommodate messages in different formats).

The ability to add such messageflow nodes would most likely be limited to certain subscribers by authorization, to prevent subscribers from causing degradation of service by expensive or damaging subscriber owned messageflows. Further, once such additional nodes are added, they  
35 remain private with respect to the subscriber who has requested such additional nodes, resulting in the ability to reuse names for such nodes between different subscribers.

## CLAIMS

1. A message broker data processing apparatus comprising:

means for receiving published messages on a topic from a publisher application;

means for processing the received messages; and

means for distributing the processed messages to subscriber applications;

wherein the apparatus further includes:

means for receiving a subscription request from a subscriber application, the subscription request including an indication of a data processing node which the subscriber application would like to add to the apparatus; and

means for, in response to the indication in the subscription request, adding the data processing node to the apparatus.

2. The apparatus of claim 1 wherein the means for distributing includes a plurality of subscription point data processing nodes, each of which distributes messages to at least one subscriber application which has previously registered a subscription request identifying a specific one of the plurality of subscription point data processing nodes with the broker apparatus.

3. The apparatus of claim 1 wherein said apparatus communicates with a subscriber application over the Internet and wherein at least one of the subscriber applications and the publisher application runs in conjunction with a World Wide Web browser.

4. A message broker data processing method comprising steps of:

receiving published messages on a topic from a publisher application;

processing the received messages via a messageflow comprising at least one data processing node; and

distributing the processed messages to subscriber applications;

wherein the method further includes steps of:

receiving a subscription request from a subscriber application, the subscription request including an indication of a data processing node which the subscriber application would like to add to the messageflow; and

5

in response to the indication in the subscription request, adding the data processing node to the messageflow.

10

5. The method of claim 4 wherein the distributing step includes using a plurality of subscription point data processing nodes, each of which distributes messages to at least one subscriber application which has previously registered a subscription request identifying a specific one of the plurality of subscription point data processing nodes.

15

6. The method of claim 4 wherein a subscription request is received from a subscriber application over the Internet and wherein at least one of the subscriber applications and the publisher application runs in conjunction with a World Wide Web browser.

20

7. A computer program product stored on a computer readable storage medium for, when run on a computer, instructing the computer to carry out the method steps recited in claim 4.



Application No: GB 9922818.1  
Claims searched: 1-7

Examiner: Mike Davis  
Date of search: 20 April 2000

**Patents Act 1977**  
**Search Report under Section 17**

**Databases searched:**

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): G4A (AFGDC, APX, AUIDB)

Int Cl (Ed.7): G06F

Other:

**Documents considered to be relevant:**

Category	Identity of document and relevant passage	Relevant to claims
X	EP 0759591 A1 (IBM) eg abstract and page 28 lines 33-57	1-7
X	WO 91/03017 A1 (MICROSOFT) eg abstract and page 5 line 23 to page 6 line 4	1,3,4,6,7
X	US 4815030 (CROSS) eg column 4 line 34 to column 6 line 5, and column 8 lines 3-27	1-7

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.